Conception et développement d'une plateforme d’hébergement d’événements

[Dossier d’analyse et de conception]

Équipe projet 20

Chef projet

- Kenzo Sechi

Membre projet

- Matteo Banroques-Mathian
- Samuel Schicke
- Toni Rey
- Maxime Rastelli
- Hashem Belal

Introduction :

Ce document est une synthèse de la conception de l'application Mont-Ski, une application lourde permettant la gestion d'un club de ski.

Nos diagrammes présentent la structure de notre application de l'inscription d'un participant à l'attribution de matériel et sa gestion, incluant également la création de groupes par niveaux et âges ainsi que la gestion d'un réseau de transports en commun acheminant les skieurs partant des villages alentour jusqu'à la station de ski.

Les diagrammes sont réalisés pour répondre à certaines problématiques de la manière la plus simple et optimisé possible, de plus, certaines fonctionnalités nécessitant initialement une base de données ont dû être réalisés de manière à ce qu'elles utilisent un stockage local ce qui nous a contraint à adapter nos cas d'utilisation ainsi que notre diagramme de classes.

Vous trouverez dans ce document l'ensemble du processus de recherche ainsi que de conception UML de l'application.

Expression des besoins et Analyse :

Description des cas d'utilisation :

Figure 1 : Diagramme des cas d’utilisation pour l’application Mont-Ski

Nom du cas d'utilisation	UC1
Libellé du cas d’utilisation	Inscrire un adhérent pour la saison
Description	*Un adhérent appelle l'organisateur pour s’inscrire à la saison de ski à venir.* L’organisateur demande à la plate-forme d’hébergement d’événement d’ajouter l’adhérent identifié par son nom, prénom, âge et niveau de ski à la liste des participants pour la saison.

Nom du cas d'utilisation	UC2
Libellé du cas d’utilisation	Inscrire un moniteur pour la saison
Description	*Un moniteur appelle l'organisateur pour s’inscrire à la saison de ski à venir.* L’organisateur demande à la plate-forme d’hébergement d’événement d’ajouter le moniteur identifié par son nom, prénom et âge à la liste des moniteurs pour la saison.

Nom du cas d'utilisation	UC3
Libellé du cas d’utilisation	Ajouter un adhérent à un groupe
Description	*Un adhérent appelle l'organisateur pour s’inscrire à un groupe de ski*. L’organisateur demande à la plate-forme d’hébergement d’événement d’ajouter un adhérent identifié par son nom prénom âge et niveau au ski dans un groupe.

Nom du cas d'utilisation	UC4
Libellé du cas d’utilisation	Ajouter un moniteur à un groupe

Description	*Un moniteur appelle l'organisateur pour s'inscrire à un groupe de ski*. L'organisateur demande à la plate-forme d'hébergement d'événement d'ajouter un moniteur identifié par son nom prénom âge dans un groupe.

Nom du cas d'utilisation	UC5
Libellé du cas d'utilisation	Former un groupe
Description	*Un moniteur /ou adhérent appelle l'organisateur pour s'inscrire à un groupe de ski cependant aucun groupe ne lui correspond(tranche d'âge et niveau*. L'organisateur demande à la plate-forme d'hébergement d'événement de créer un nouveau groupe identifié par un nom, niveau et une tranche d'âge.

Nom du cas d'utilisation	UC6
Libellé du cas d'utilisation	Désinscrire un adhérent
Description	*Suite à un accident sur les pistes(blessures etc…) L'adhérent appelle l'organisateur pour se désinscrire d'un groupe de ski.* L'organisateur demande à la plate-forme d'hébergement d'événement d'enlever l'adhérent du groupe.

Nom du cas d'utilisation	UC7
Libellé du cas d'utilisation	Désinscrire un moniteur
Description	*Suite à un accident sur les pistes(blessures etc…) Le moniteur appelle l'organisateur pour se désinscrire d'un groupe de ski.* L'organisateur demande à la plate-forme d'hébergement

	d'événement d'enlever le moniteur du groupe.

Nom du cas d'utilisation	UC8
Libellé du cas d'utilisation	Consulter les inscriptions.
Description	*Le membre d'un groupe appelle l'organisateur pour vérifier ses informations et le groupe dans lequel il est inscrit.* L'organisateur demande à la plate-forme d'hébergement d'événement d'afficher la participation.

Nom du cas d'utilisation	UC9
Libellé du cas d'utilisation	Gérer le matériel de ski
Description	*Un membre du club appelle l'organisateur pour la location de matériel de ski.* L'organisateur demande à la plate-forme d'hébergement d'événement de réserver le matériel de ski demandé.

Nom du cas d'utilisation	UC10
Libellé du cas d'utilisation	Planifier les transports
Description	*Le membre d'un groupe appelle l'organisateur pour prévenir qu'il a besoin d'être transporté pour aller au point de rendez-vous.* L'organisateur demande à la plate-forme d'hébergement d'événement de réserver une place dans les transports du club.

Explication des cas d'utilisation en fonction des scénarios alternatifs:

Contexte : l'utilisateur ne peut pas interagir avec la plateforme, il doit passer par l'organisateur pour le faire.

Nom du scénario nominal	UC1.nominal
Description	1. L'adhérent souhaite s'inscrire à la saison de ski à venir. 2. L'organisateur demande à l'adhérent de fournir son nom, prénom, âge et niveau au ski. 3. L'adhérent fournit ces quatre informations. 4. L'organisateur demande à la plate-forme d'ajouter un adhérent à la liste des participants pour la saison. 5. La plate-forme ajoute l'adhérent à la liste des participants pour la saison.

Nom du scénario nominal	UC1.adhérent déjà inscrit
Description	À l'étape 4, si l'adhérent donne un nom et prénom déjà présent dans la liste des participants, une erreur apparaît.

Nom du scénario nominal	UC1.taille maximale
Description	À l'étape 4, si l'ajout de l'adhérent surpasse le nombre maximum de groupes, une erreur apparaitra.

Nom du scénario nominal	UC2.nominal
Description	1. Le moniteur souhaite s'inscrire à la saison de ski à venir. 2. L'organisateur demande au moniteur de fournir son nom, prénom et âge. 3. Le moniteur fournit ces trois informations.

	4. L’organisateur demande à la plate-forme d’ajouter un moniteur à la liste des moniteurs pour la saison. 5. La plate-forme ajoute le moniteur à la liste des moniteurs pour la saison.

Nom du scénario nominal	UC2.moniteur déjà inscrit
Description	À l’étape 4, si le moniteur donne un nom et prénom déjà présent dans la liste des moniteurs, une erreur apparaît.

Nom du scénario nominal	UC2.taille maximale
Description	À l’étape 4, si l’ajout du moniteur surpasse le nombre maximum de groupes, une erreur apparaitra.

Nom du scénario nominal	UC3.nominal
Description	1. L'adhérent souhaite s’inscrire à une sortie de ski. 2. L’organisateur demande à l'adhérent de fournir son nom, prénom, âge et niveau au ski. 3. L'adhérent fournit ces quatre informations. 4. L’organisateur demande à la plate-forme d’ajouter un adhérent à un groupe. 5. La plate-forme ajoute l’adhérent à un groupe.

Nom du scénario nominal	UC3.adhérent déjà inscrit
Description	À l’étape 4, si l’adhérent donne un nom et prénom déjà présent dans un groupe, une erreur apparaît.

Nom du scénario nominal	UC3.moniteur insuffisant

Description	À l'étape 4, si l'ajout de l'adhérent entraîne le non-respect du taux d'encadrement par moniteur, une erreur apparaitra.

Nom du scénario nominal	UC3.taille maximale
Description	À l'étape 4, si l'ajout de l'adhérent dépasse la taille du nombre de personnes autorisées, une erreur apparaitra.

Nom du scénario nominal	UC4.nominal
Description	1. Le moniteur souhaite s'inscrire à une sortie de ski. 2. L'organisateur demande au moniteur de fournir son nom, prénom, âge. 3. Le moniteur fournit ces trois informations. 4. L'organisateur demande à la plate-forme d'ajouter un moniteur à un groupe. 5. La plate-forme ajoute le moniteur à un groupe.

Nom du scénario nominal	UC4.moniteur déjà inscrit
Description	À l'étape 4, si le moniteur donne un nom et prénom déjà présent dans un groupe, une erreur apparaitra.

Nom du scénario nominal	UC4.taille maximale
Description	À l'étape 4, si l'ajout du moniteur surpasse la capacité maximale du groupe, une erreur apparaitra.

Nom du scénario nominal	UC5.nominal
Description	1. Un moniteur ou un adhérent souhaite s'inscrire à un groupe de ski, mais aucun groupe ne lui correspond. 2. L'organisateur demande au moniteur ou à l'adhérent de fournir son nom,

	prénom, âge et niveau au ski. 3. Le moniteur ou l'adhérent fournit ces informations. 4. L'organisateur demande à la plate-forme de créer un nouveau groupe avec les informations fournies. 5. La plate-forme crée un nouveau groupe.

Nom du scénario nominal	UC5.groupe existant
Description	À l'étape 4, si un groupe avec le même nom, tranche d'âge et niveau existe déjà, une erreur apparaît.

Nom du scénario nominal	UC6.nominal
Description	1. Le membre d'un groupe souhaite vérifier ses informations d'inscription. 2. L'organisateur demande au membre de fournir son nom, prénom et groupe de ski. 3. Le membre fournit ces informations. 4. L'organisateur demande à la plate-forme d'afficher les informations d'inscription du membre. 5. La plate-forme affiche les informations d'inscription du membre.

Nom du scénario nominal	UC6.adhérent non inscrit
Description	À l'étape 4, si l'adhérent ne figure pas dans le groupe, une erreur apparaît.

Nom du scénario nominal	UC7.nominal
Description	1. Le moniteur souhaite se désinscrire d'un groupe de ski suite à un accident. 2. L'organisateur demande au moniteur de fournir son nom,

	prénom et groupe de ski. 3. Le moniteur fournit ces informations. 4. L'organisateur demande à la plate-forme de désinscrire le moniteur du groupe. 5. La plate-forme enlève le moniteur du groupe.

Nom du scénario nominal	UC7.moniteur non inscrit
Description	À l'étape 4, si le moniteur ne figure pas dans le groupe, une erreur apparaît.

Nom du scénario nominal	UC8.nominal
Description	1. Le membre d'un groupe souhaite vérifier ses informations d'inscription. 2. L'organisateur demande au membre de fournir son nom, prénom et groupe de ski. 3. Le membre fournit ces informations. 4. L'organisateur demande à la plate-forme d'afficher les informations d'inscription du membre. 5. La plate-forme affiche les informations d'inscription du membre.

Nom du scénario nominal	UC8.membre non inscrit
Description	À l'étape 4, si le membre ne figure pas dans le groupe, une erreur apparaît.

Nom du scénario nominal	UC9.nominal
Description	1. Le membre souhaite faire la location de matériel. 2. L'organisateur demande au membre de fournir son nom, prénom, groupe de ski, taille, et poid. 3. Le membre fournit ces informations. 4. L'organisateur demande à la plate-forme de réserver le matériel. 5. La plate-forme affiche le matériel loué.

Nom du scénario nominal	UC9.matériel indisponible
Description	À l'étape 4, Si le matériel demandé est en rupture de stock, une erreur apparaît.

Nom du scénario nominal	UC10.nominal
Description	1. Le membre à besoin d'être transporté. 2. L'organisateur demande au membre de fournir son nom, prénom, groupe de ski, taille, et poid. 3. Le membre fournit ces informations. 4. L'organisateur demande à la plate-forme de réserver une place dans les transports du club. 5. La plate-forme affiche la réservation.

Nom du scénario nominal	UC10.place non disponible

Description	À l’étape 4, si la place dans le transport n'est pas disponible, une erreur apparaît.

Priorisation des cas d’utilisation :

Priorité	Cas d'utilisation	Justification
1	UC1	Ce cas d'utilisation est essentiel car il permet d'assurer la participation des adhérents pour la saison de ski à venir, garantissant ainsi le fonctionnement global de l'organisation.
	UC2	Ce cas d'utilisation est essentiel car il permet d'assurer la participation des moniteurs pour la saison de ski à venir, garantissant ainsi le fonctionnement global de l'organisation.
	UC3	Ces cas d'utilisation sont une nécessité à l'organisation des sorties de ski, en assignant les adhérents aux groupes appropriés, cela assure le maintien d’une bonne organisation.
	UC4	Ces cas d'utilisation sont une nécessité à l'organisation des sorties de ski, en assignant les moniteurs aux groupes appropriés, cela assure le maintien d’une bonne organisation.
	UC5	Former de nouveaux groupes en fonction des tranches d'âge et des niveaux de compétence est obligatoire pour pouvoir organiser les sorties de ski.
2	UC6	La désinscription des adhérents suite à des incidents est utile pour une gestion efficace des groupes de ski et ne pas créer des nouveaux groupes pour rien. Ce use-case est bien moins important que les fonctionnalités.
	UC7	La désinscription des moniteurs suite à des incidents est utile pour une gestion efficace des groupes de ski et ne pas créer des nouveaux groupes pour rien. Ce use-case est bien moins important que les fonctionnalités.
	UC8	Bien que la consultation des inscriptions soit utile pour les participants, elle est moins critique que les autres activités liées à la

		gestion des inscriptions et peut être considérée comme une tâche administrative secondaire.
3	UC9	Bien que la location de matériel soit importante pour assurer le bon déroulement des activités de ski, elles sont plus secondaires et peuvent être traitées en parallèle des autres priorités.
	UC10	Bien que les transports soient importants pour assurer le bon déroulement des activités de ski, elles sont plus secondaires et peuvent être traitées en parallèle des autres priorités.

Architecture proposée :

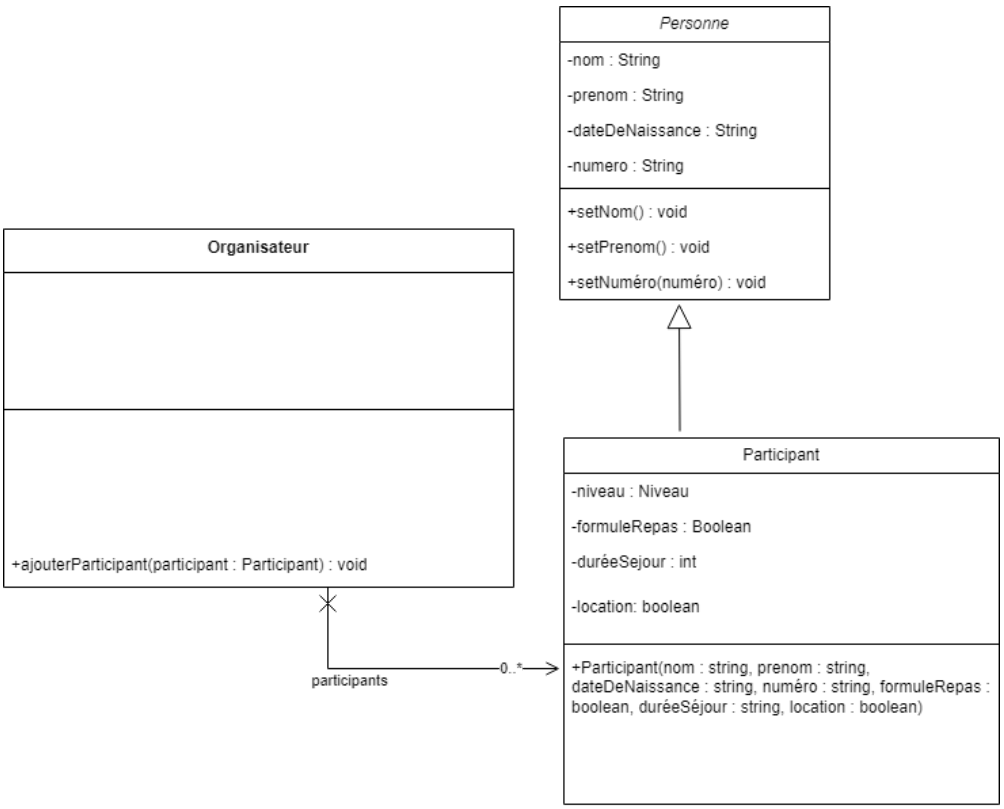
Exemple d'utilisation :

Conception détaillée

Architecture détaillée

Diagramme de classe détaillée métier du premier cas d'utilisation :

Le diagramme représente l'inscription d'un participant pour la saison de ski.

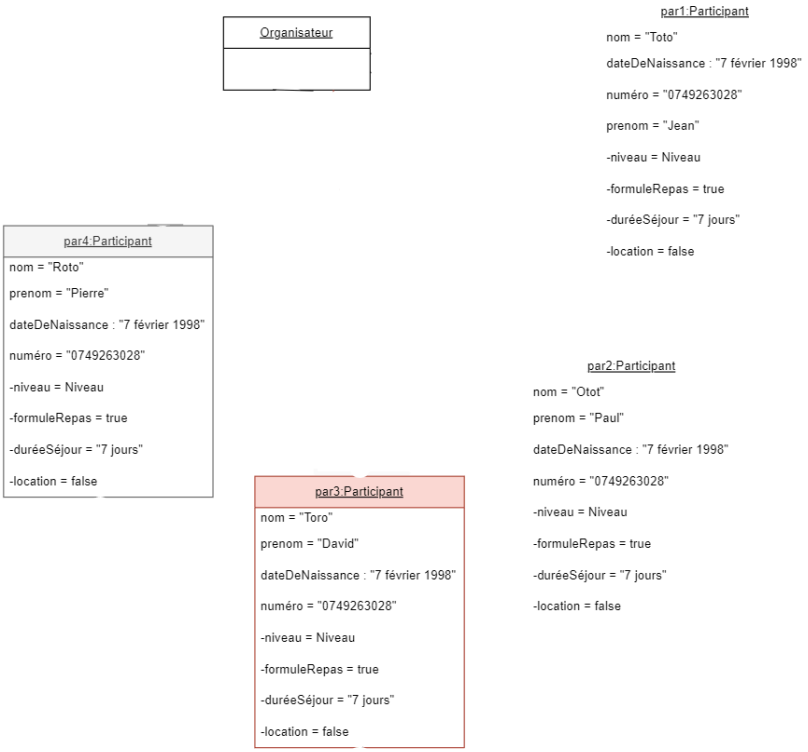


Détail des scénarios

Conclusion

Diagrammes d’objets :

DIAGRAMME D’OBJET UC1/UC6/UC8 (Inscrire un adhérent pour la saison) :
L’organisateur ajoute un nouvel adhérent à la liste des adhérents ou le supprime.
état du système avant réalisation :



état du système après réalisation :

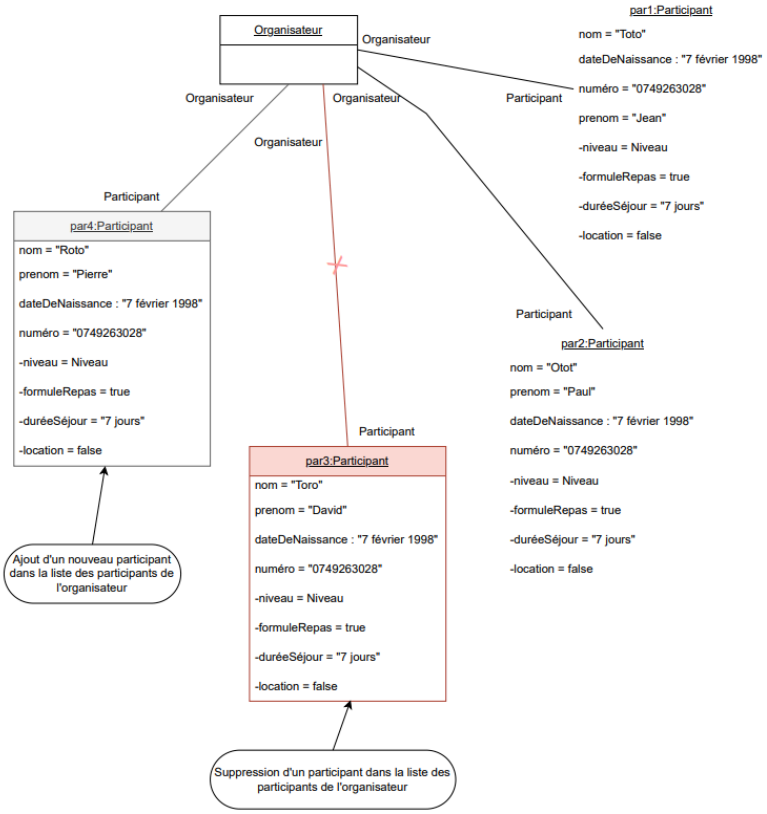
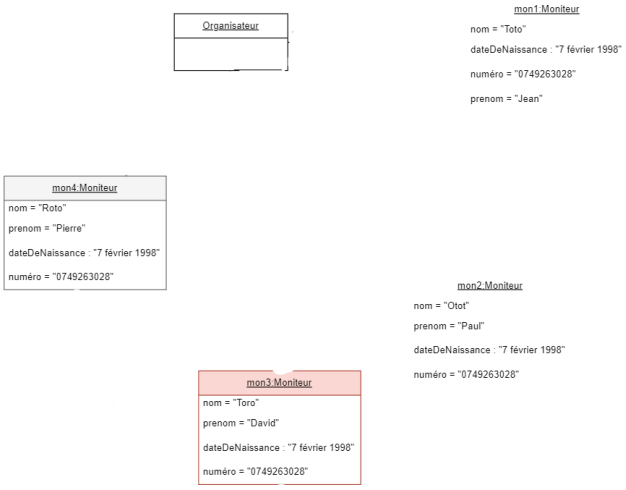


DIAGRAMME D'OBJET UC2/UC7 (Inscrire un moniteur pour la saison) :
L'organisateur ajoute un nouveau moniteur à la liste des moniteurs ou le supprime.
état du système avant réalisation :



état du système après réalisation :

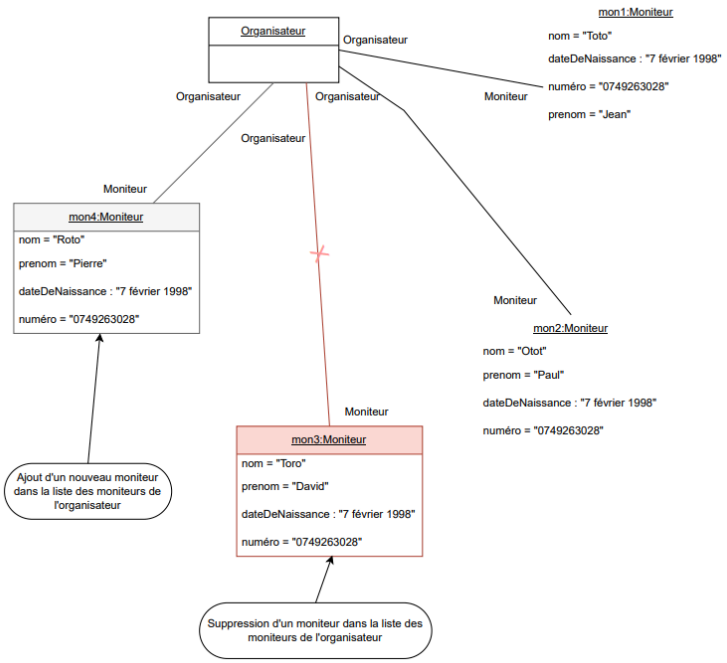
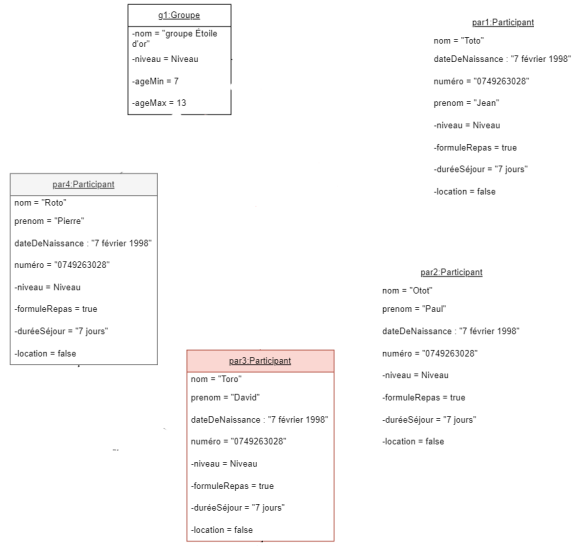


DIAGRAMME D'OBJET UC3 (Inscrire un adhérent à un groupe pour la saison) :
L'organisateur ajoute un adhérent au groupe si il y a moins de 15 personnes dans le groupe.
état du système avant réalisation :



état du système après réalisation :

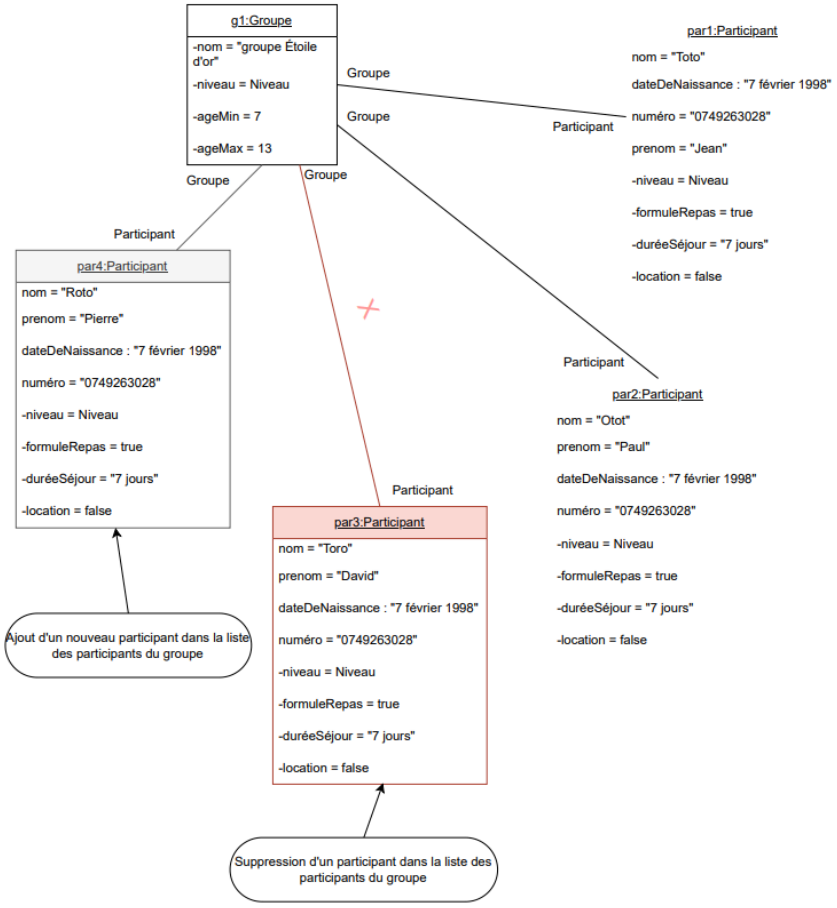
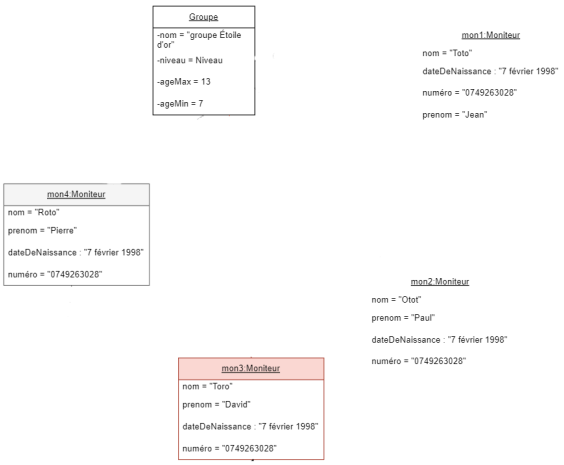


DIAGRAMME D'OBJET UC4 (Inscrire un moniteur à un groupe pour la saison) :
L'organisateur ajoute un moniteur au groupe avec pour maximum 2 moniteurs par groupes.
état du système avant réalisation :



état du système après réalisation :

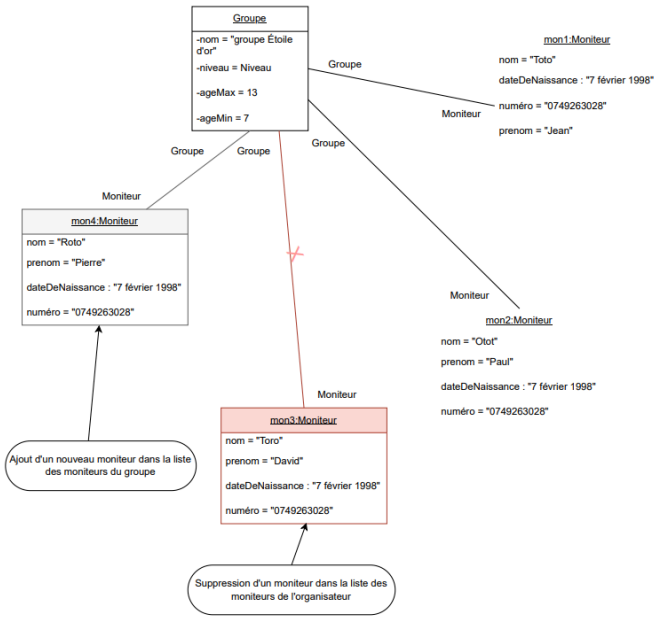
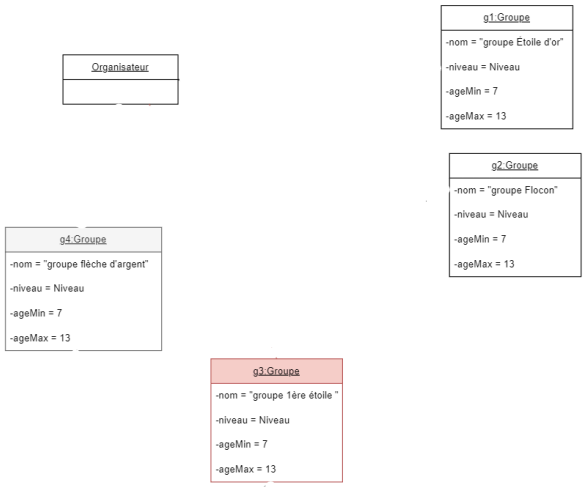


DIAGRAMME D'OBJET UC5 (créer un groupe pour la saison) :
L'organisateur créer un nouveau groupe qui s'ajoute à la liste des groupes.
état du système avant réalisation :



état du système après réalisation :

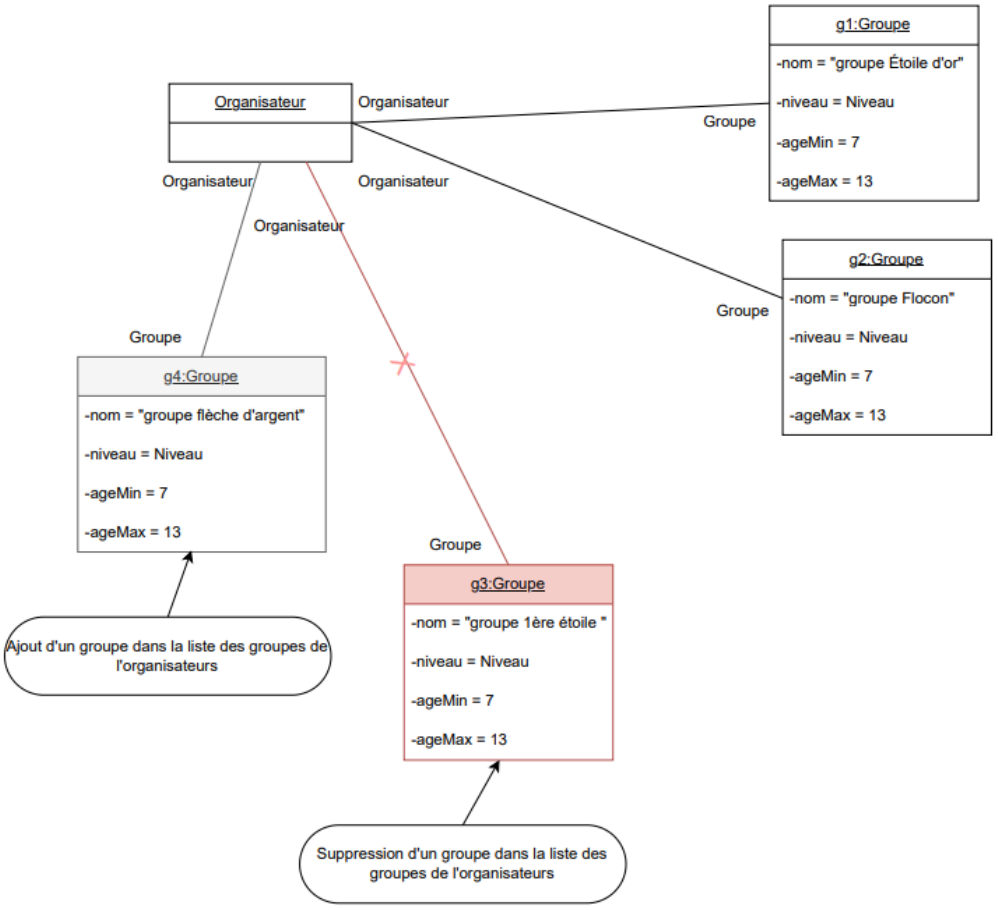


DIAGRAMME D'OBJET UC9 (Gérer le matériel de ski pour la saison) :
L'organisateur attribue un matériel en location à un adhérent.
état du système avant réalisation :

état du système après réalisation :

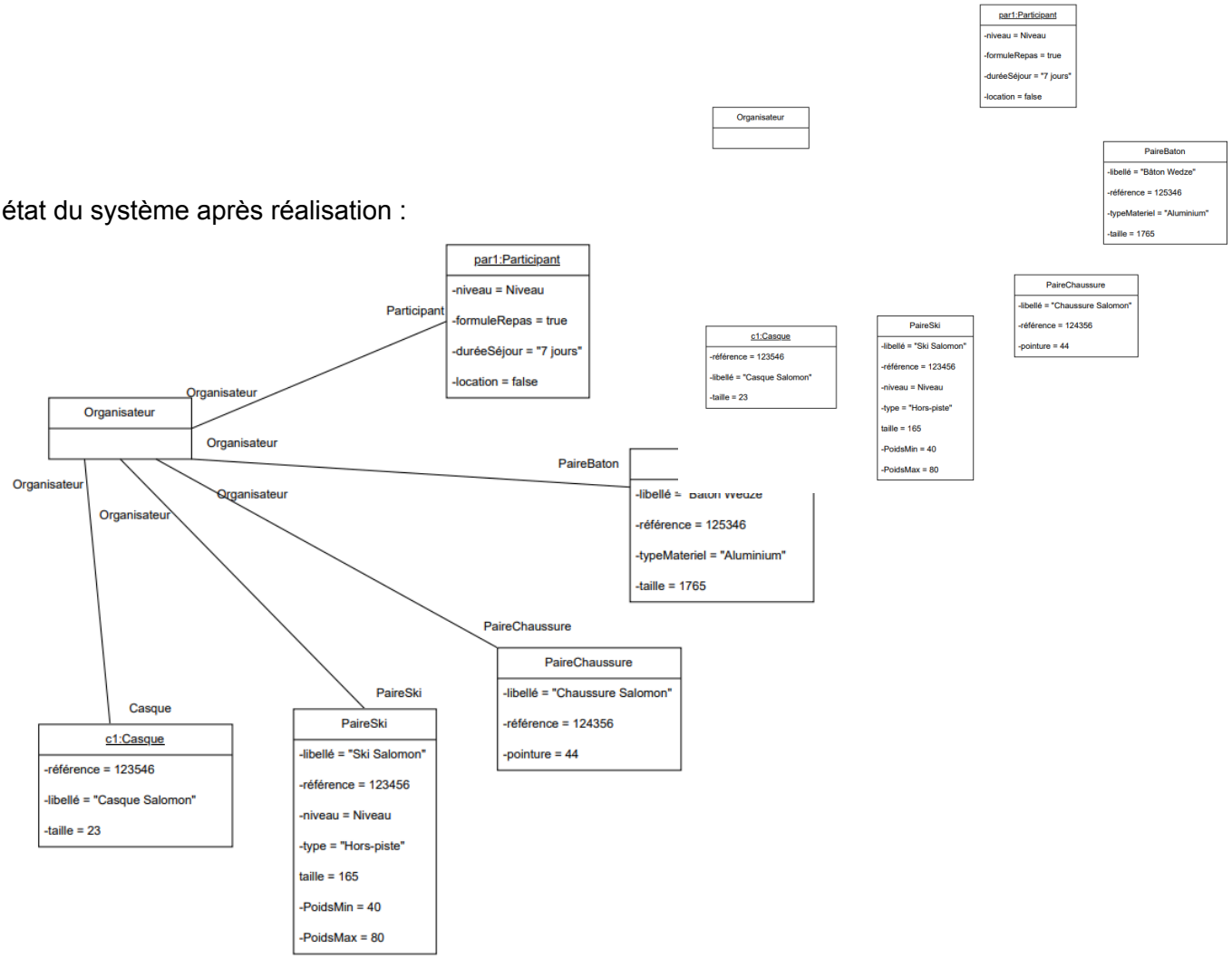
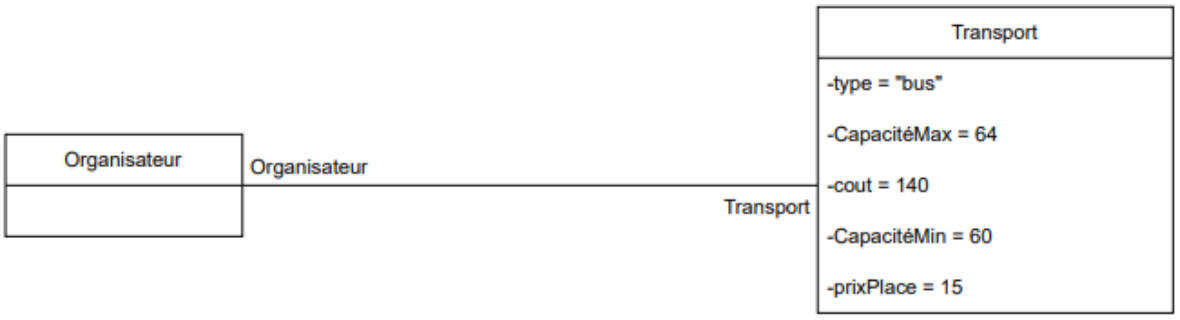


DIAGRAMME D'OBJET UC10 (Gérer le transport des membres du club de ski pour la saison) :
L'organisateur ajoute un transport pour récupérer les adhérents.

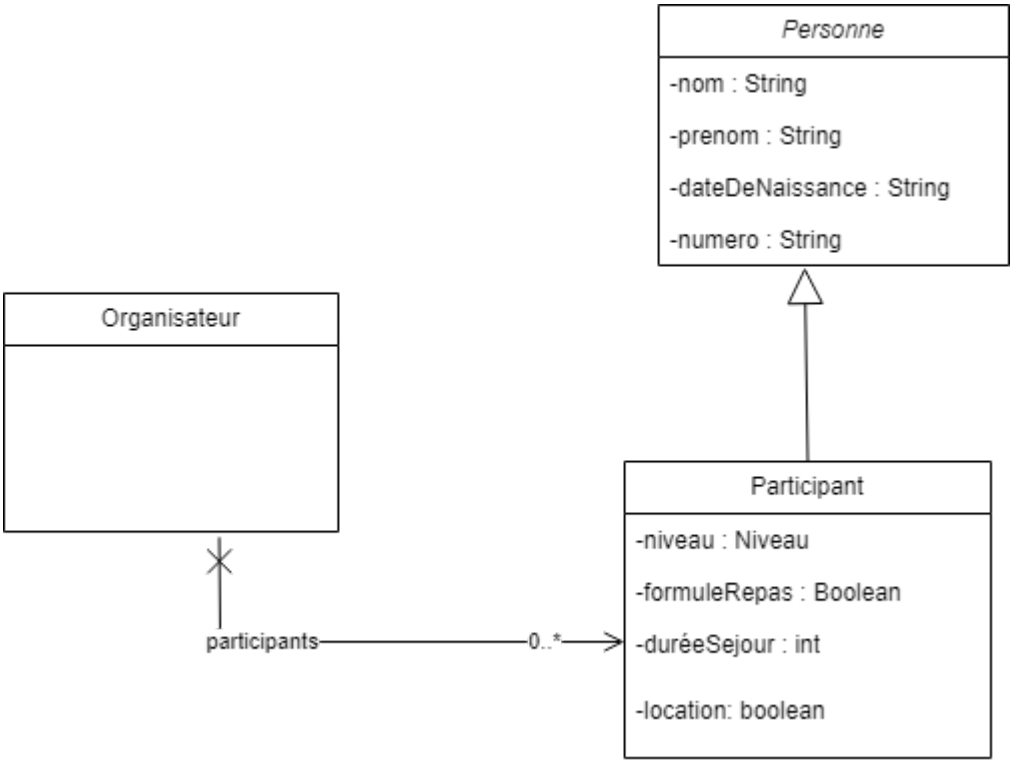
état du système avant réalisation :

état du système après réalisation :



diagrammes métier :

diagramme métier uc1:



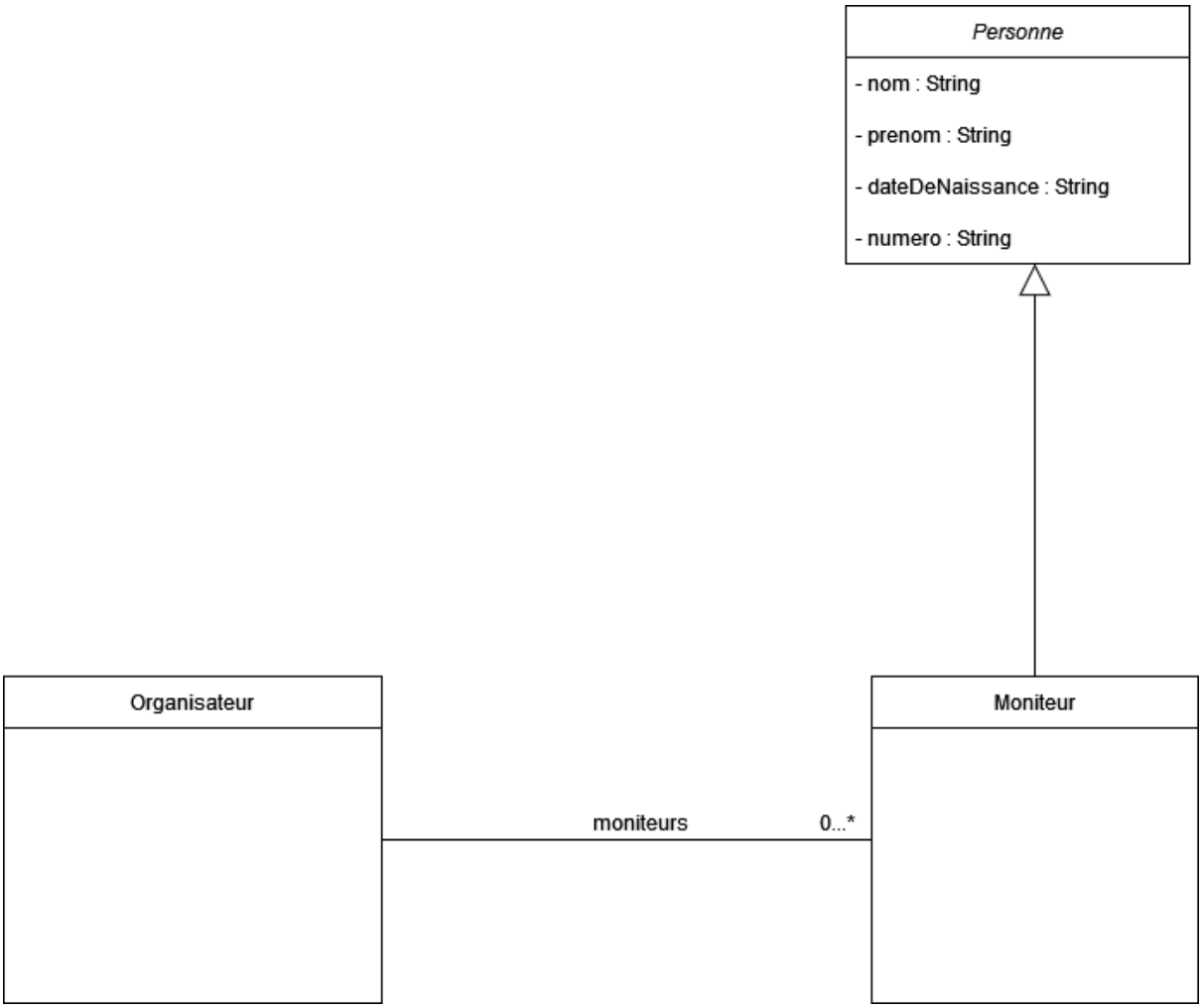
Ce diagramme représente le cas d'utilisation numéro 1 (Inscrire un adhérent pour la saison) en fonction des différentes classes qu'il utilise .

Nous pouvons distinguer trois classes : Personne, Participants, Organisateur

- une cardinalité 0..* d'Organisateur à Participant signifiant qu'Organisateur est composé d'une liste de participants.

La classe Participant est une classe fille de Personne.

diagramme métier uc2:

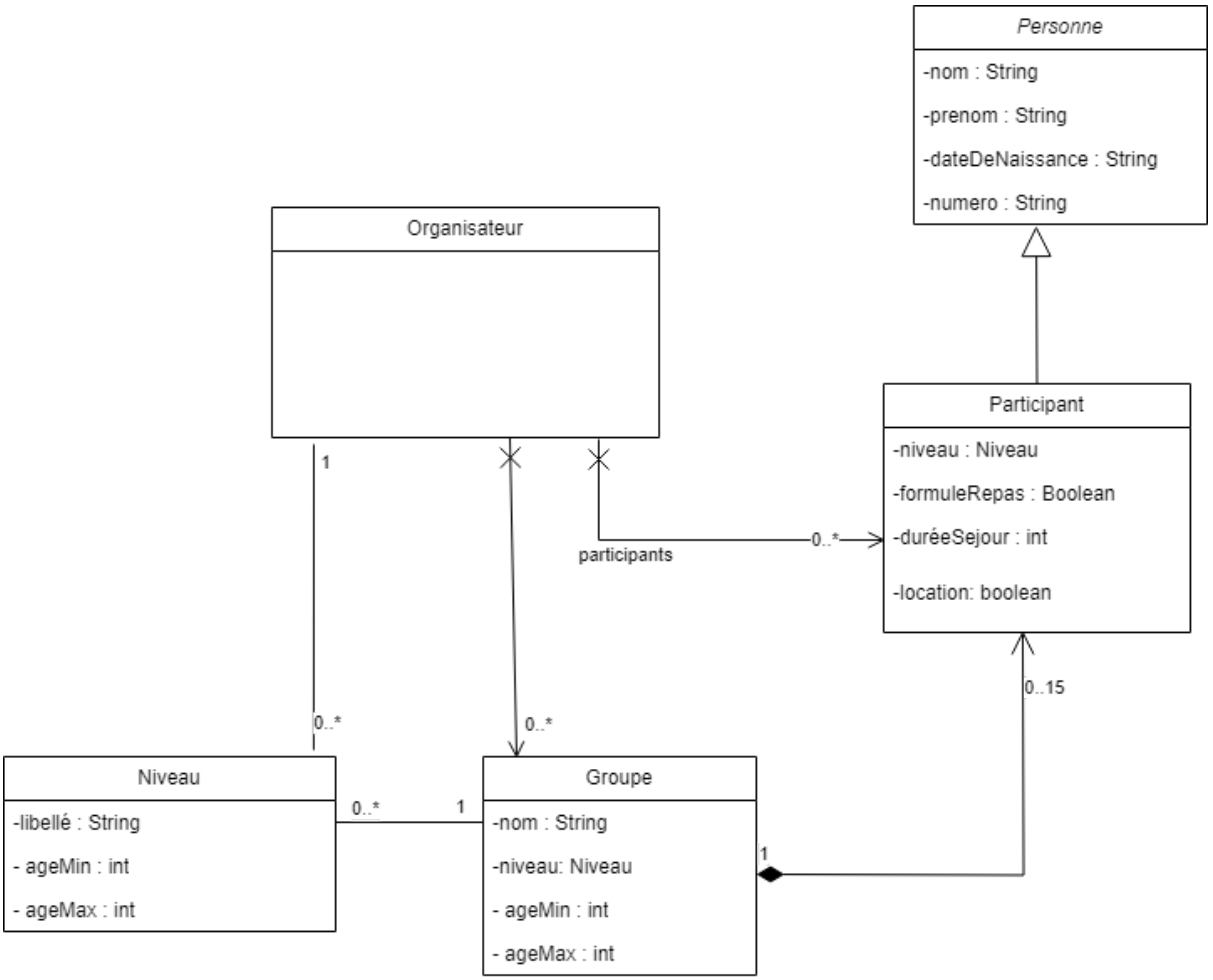


Ce diagramme représente le cas d'utilisation numéro 2 (Inscrire un moniteur pour la saison) en fonction des différentes classes qu'il utilise . Nous pouvons distinguer trois classes : Personne, Moniteurs et Organisateur.

Nous avons aussi une cardinalité 0..* d'Organisateur à Moniteur signifiant qu'Organisateur est composé d'une liste de moniteurs . La classe Participant est une classe fille symbolisée par la flèche pointant vers Personne sa classe mère.

L'ajout d'un nouveau moniteur est symbolisé par une liaison entre Organisateur vers Participant.

diagramme uc3 :



Ce diagramme représente le cas d'utilisation 3 (Ajouter un adhérent à un groupe) en fonction des différentes classes qu'il utilise .

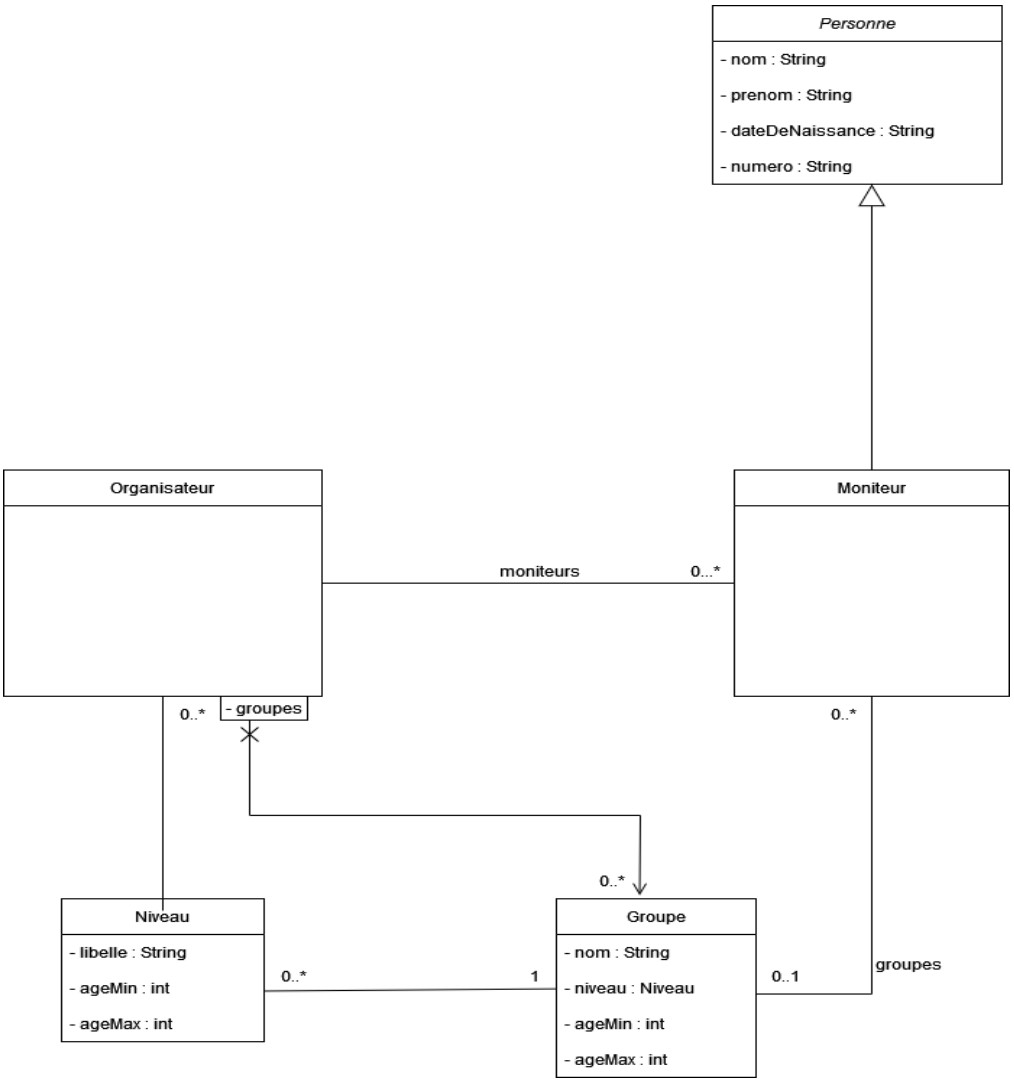
Nous pouvons distinguer cinq classes : Personne, Participants, Organisateur,Niveau et Groupe.

Nous avons :

- une cardinalité 0..* d'Organisateur à Participant signifiant qu'Organisateur est composé d'une liste de participants.
- un lien de composition entre Groupe et Participant signifiant qu'un groupe est composé de 0 à 15 participants.
- un lien entre Niveau et Groupe qui nous dit que un niveau a de 0 à plusieurs groupes et qu'un groupe a seulement 1 niveau
- Il y a également une cardinalité entre Niveau et organisateur signifiant qu'un niveau à qu'un organisateur et qu'un organisateur a une liste de o à n niveaux

La classe Participant est toujours fille de Personne selon les explications précédentes.L'ajout d'un nouveau moniteur est symbolisé par une liaison entre Organisateur vers Participant.

diagramme métier UC4 :



Ce diagramme représente le cas d'utilisation 4 (Ajouter un moniteur à un groupe) en fonction des différentes classes qu'il utilise.

Nous pouvons distinguer cinq classes : Personne, Moniteur, Groupe, Niveau et Organisateur.

-Un moniteur est une personne, cela est représenté par la flèche d'héritage de la classe moniteur jusqu'à Personne .

-Nous avons aussi une cardinalité 0..* d'Organisateur à Moniteur signifiant qu'Organisateur est composé d'une liste de participants. La classe Participant est une classe fille symbolisée par la flèche pointant vers Personne sa classe mère.

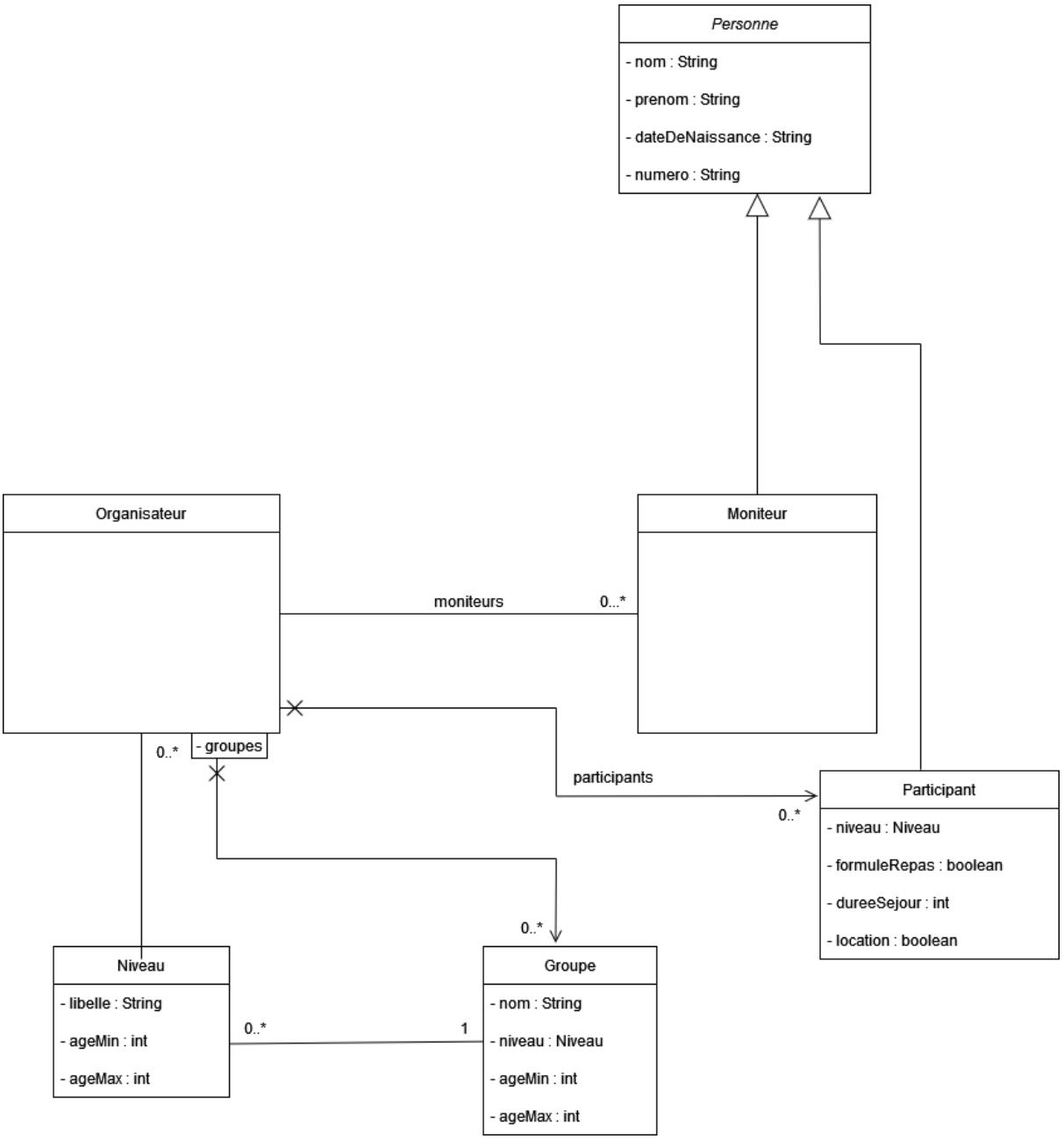
-Le lien entre moniteur et groupe veut dire qu’un moniteur peut appartenir à un groupe ou aucun(0..1) et un groupe contient aucun ou plusieurs moniteurs.

-La cardinalité entre niveau et groupe 0..* signifie qu’un niveau peut être attribué à plusieurs groupes tandis que la cardinalité à 1 exprime le fait qu’un groupe contient seulement un niveau.

-Le lien entre organisateur et niveau accompagné par la cardinalité 0..* exprime la possibilité pour l’organisateur de créer 0 ou plusieurs niveaux.

L'ajout d'un nouveau Moniteur est symbolisé par une liaison entre Organisateur vers Groupe.

UC5 :


Personne
- nom : String
- prenom : String
- dateDeNaissance : String
- numero : String
Organisateur
Moniteur
moniteurs
0...*
0..*
- groupes
participants
0..*
Participant
- niveau : Niveau
- formuleRepas : boolean
- dureeSejour : int
- location : boolean
0..*
Niveau
- libelle : String
- ageMin : int
- ageMax : int
0..*
1
Groupe
- nom : String
- niveau : Niveau
- ageMin : int
- ageMax : int

Ce diagramme représente le cas d'utilisation 5 (Former un groupe) en fonction des différentes classes qu'il utilise .

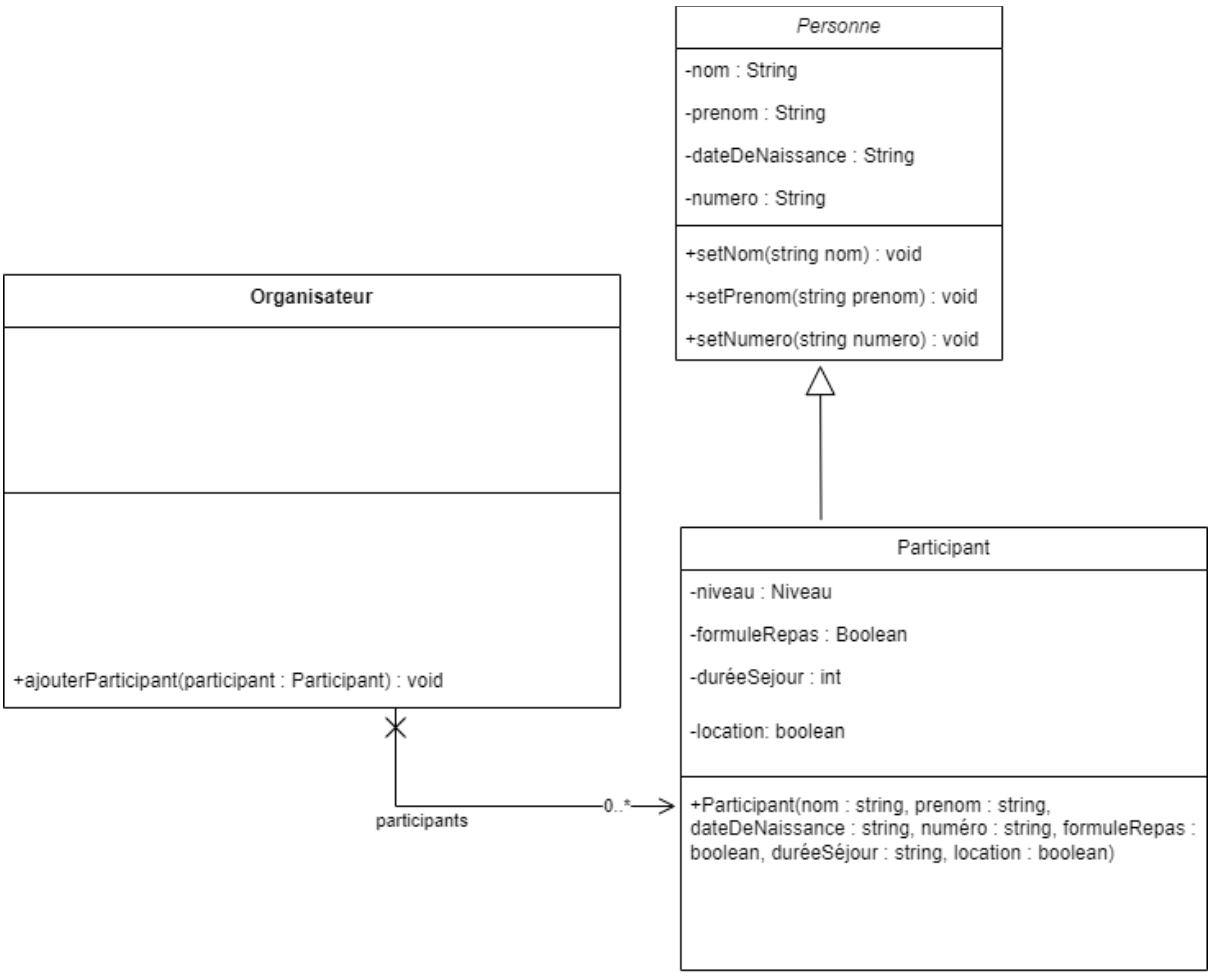
Nous pouvons distinguer six classes : Personne, Participants, Organisateur,Niveau, Groupe et Moniteur.

Les cardinalités et heritages ont été expliquées précédemment.

diagrammes métier complet :

Les diagrammes de métier complet sont des diagrammes de métier avec l'ajout de fonctions dans les classes.

uc1 complet :

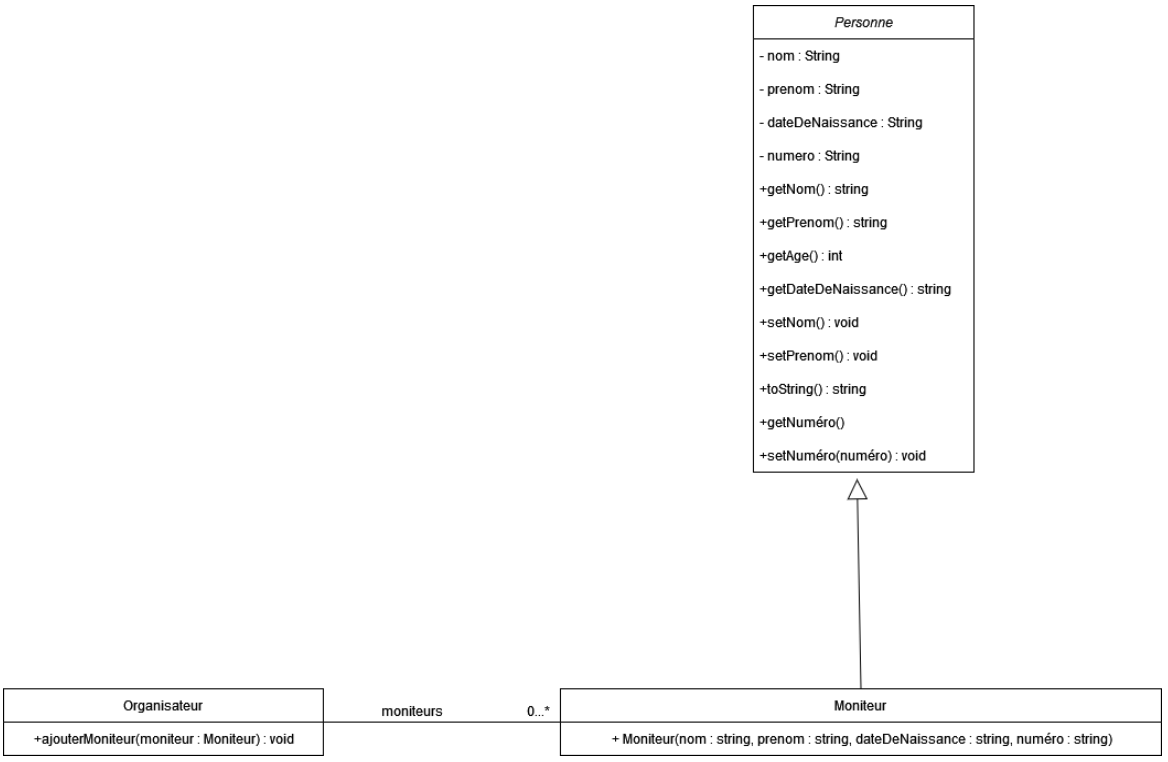


Ce diagramme représente le cas d'utilisation numéro 1 complet (Inscrire un adhérent pour la saison)

La grande différence avec le diagramme de métier complet et l'ajout de la classe:

public : ajouterPatricipant(participant: Participant) : void

uc2 complet



classe personne rajouter bar de séparation

Ce diagramme représente le cas d'utilisation numéro 2 complet (Inscrire un moniteur pour la saison). La grande différence avec le diagramme de métier complet et l'ajout de la classe:

dans Organisateur

public : ajouterMoniteur(moniteur : Moniteur) : void

dans Moniteur

public : Moniteur(nom : string, prenom :string, dateDeNaissance: string, numero:string)

dans Personne

public : getNom() : string

public : getPrenom() : string

public : gelAge() : int

public : getDateDeNaissance() : string

public : setNom(): void

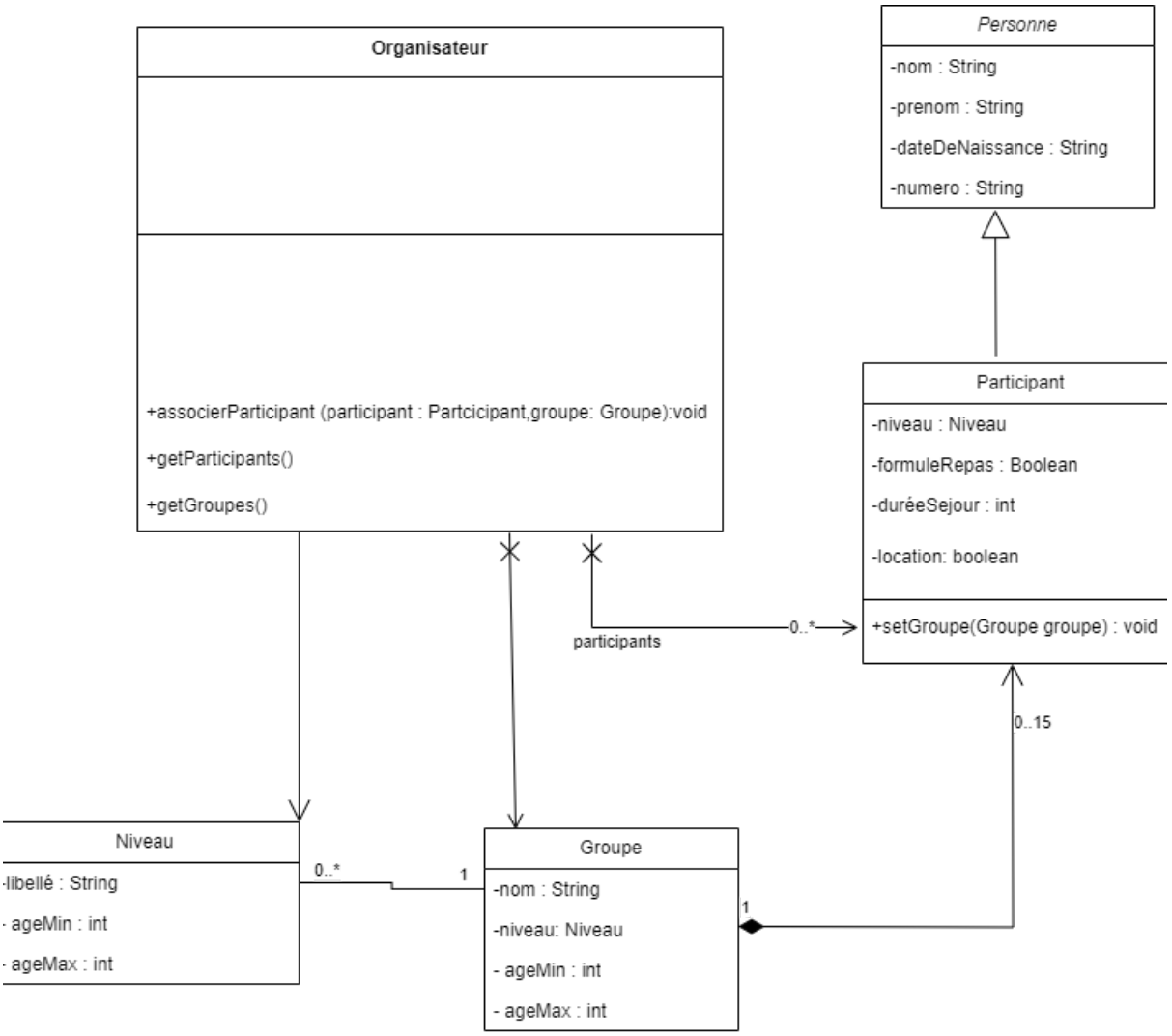
public : setPrenom() : void

public : toString() : string

public : getNuméro()

public : setNuméro(numéro) : void

uc3 compet:



Ce diagramme représente le cas d'utilisation 3 Complet (Ajouter un adhérent à un groupe) en fonction des différentes classes qu'il utilise .

la seule différence avec l'anciens est l'ajout des trois méthode dans la classe Organisateur :

dans Organisateur

public : associerParticipant (participant : Participant groupe: Groupe):void

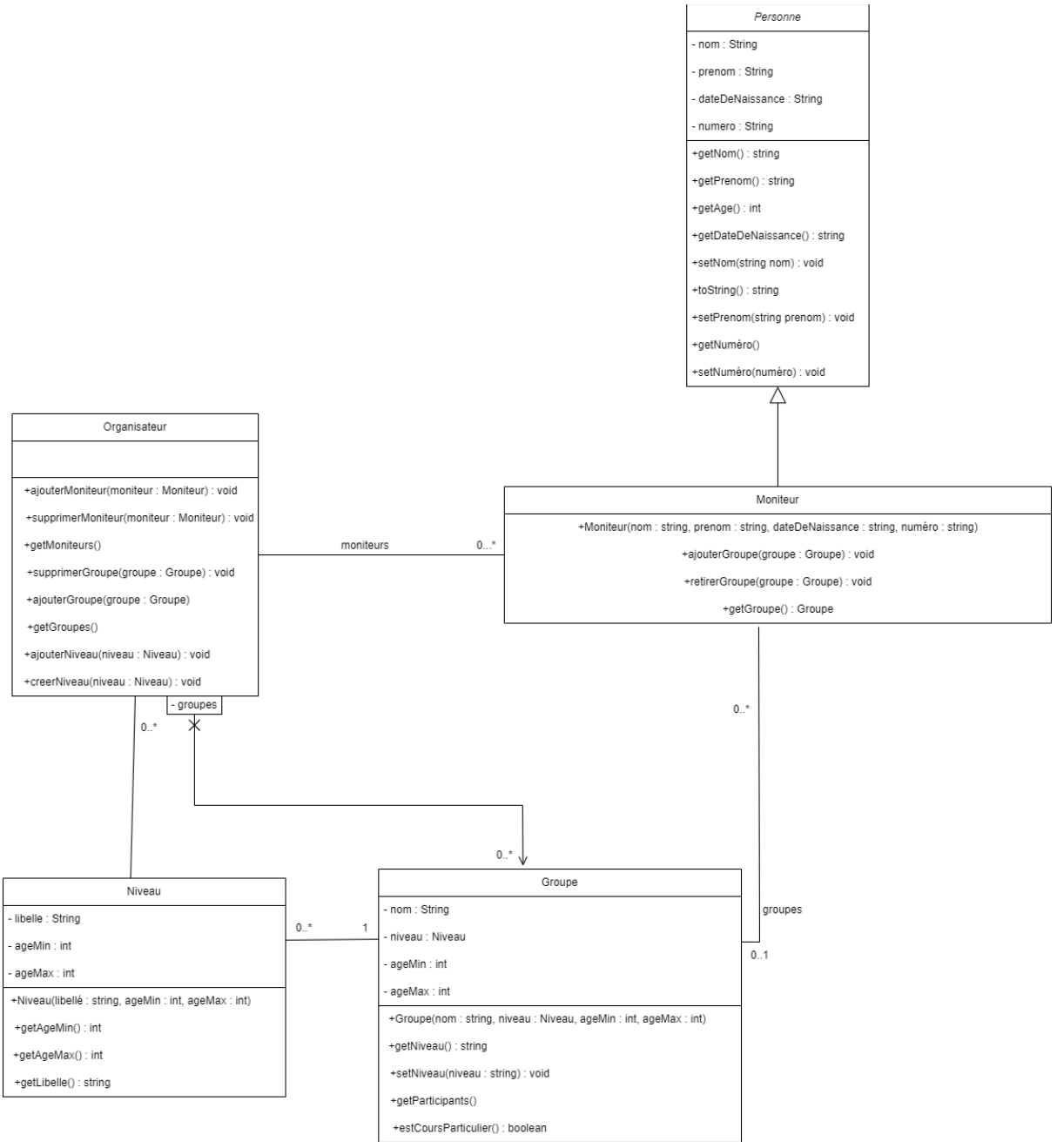
public : getParticipants()

public : getGroupes()

dans Participant

public : setGroupe(groupe groupe) : void

uc4 complet :



Ce diagramme représente le cas d'utilisation 4 Complet (Ajouter un moniteur à un groupe) en fonction des différentes classes qu'il utilise .

La seule différence avec l'anciens est l'ajout des méthode dans les classe :

dans Organisateur

public : ajouterMoniteur (moniteur Moniteur) void

public : supprimerMoniteur (moniteur: Moniteur) : void

public : getMoniteurs

public : supprimer Groupe(groupe Groupe) : void

public : ajouterGroupe(groupe : Groupe)

public : getGroupes ()

public : ajouterNiveau(niveau : Niveau) : void

public : creerNiveau(niveau : Niveau): void

dans personne

public : getNom() : string

public : getPrenom( ) : string

public : getAge() : int

public : getDateDeNaissance() : string

public : setNom(string nom) : void

public : toString() : string

public : setPrenom(string prenom) : void

public : getNumero()

public : setNuméro(numéro) : void

dans Niveau

public : Niveau(libellé :string, ageMin : int, ageMax : int)

public : getAgeMin(): int

public : getAgeMax() : int

public : getLibelle() : string

dans Groupe

public : Groupe(nom string, niveau: Niveau, ageMin int, ageMax: int)

public : getNiveau() : string

public : setNiveau(niveau string) : void

public : getParticipant()

public : estCoursParticulier() : boolean

dans Moniteur

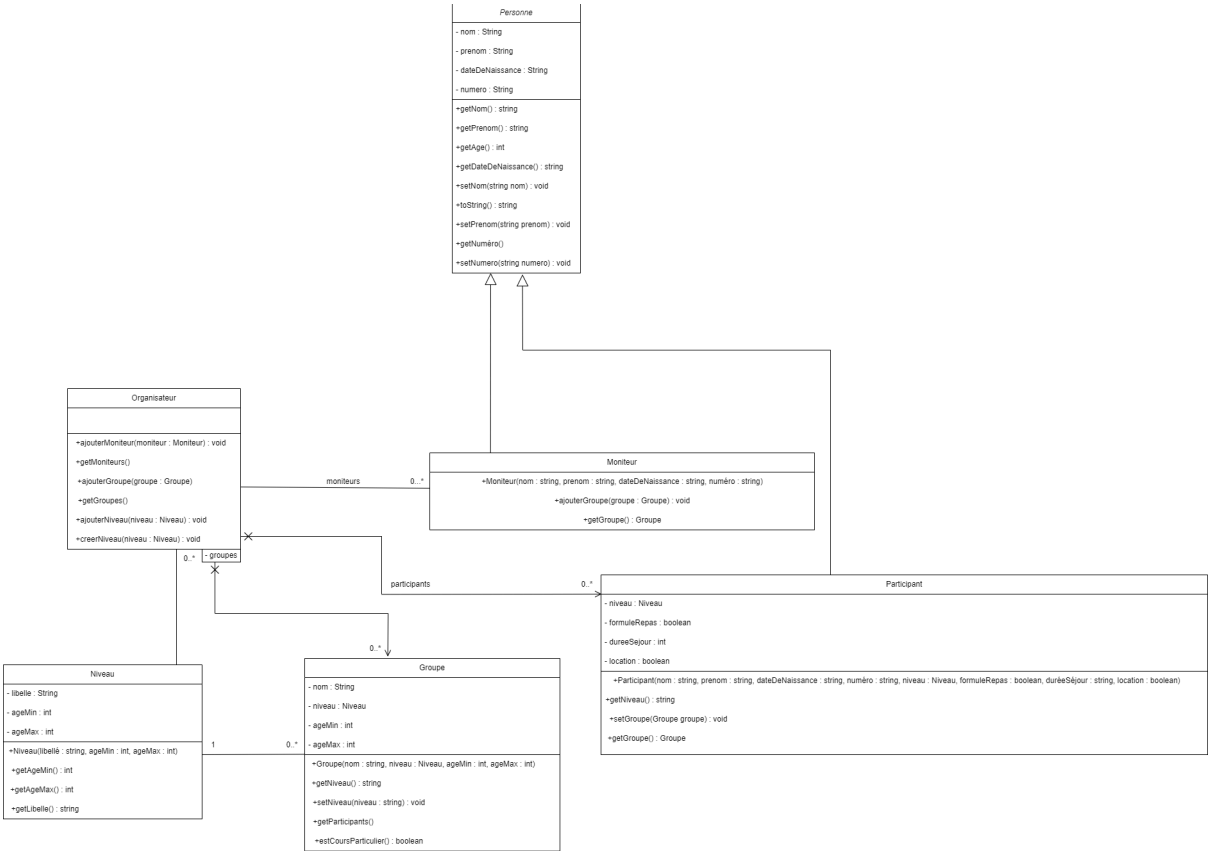
public : Moniteur(nom: string, prenom : string, dateDeNaissance: string, numéro: string)

public : ajouterGroupe(groupe : Groupe) : void

public : retirer Groupe(groupe : Groupe) : void

public : getGroupe() : Groupe

uc5 complet :



Ce diagramme représente le cas d'utilisation 5 Complet (former un groupe) en fonction des différentes classes qu'il utilise .

La seule différence avec l'anciens est l'ajout des méthode dans les classe :

-Organisateur:

public : ajouterMoniteur(Moniteur moniteur) : void

public : getMoniteurs() : HashMap<Moniteur,Moniteur>

public : ajouterParticipant(Participant participant) : void

public : getGroupes() : HashMap<Groupe,Groupe>

public : ajouterNiveau(Niveau niveau) : void

public : creerNiveau(Niveau niveau) : void

-Niveau:

public : Niveau(string libelle, int ageMin, int ageMax) : void

public : getAgeMin() : int

public : getAgeMax() : int

public : getLibelle() : string

-Groupe:

public : getNiveau() : Niveau

public : setNiveau(string niveau) : void

public : getParticipants() : HashMap<Participant,Participant>

public : estCoursParticulier() : boolean

-Moniteur:

public : Moniteur(string nom, string prenom, string dateDeNaissance, string numero) : Moniteur

public : ajouterGroupe(Groupe groupe) : void

public : getGroupes() : HashMap<Groupe,Groupe>

-Paticipant:

public : Participant(string nom, string prenom, string dateDeNaissance, string numéro, Niveau niveau, boolean formuleRepas, string duréeSéjour, boolean location) : Participant

public : getNiveau() : Niveau

public : setGroupe(Groupe groupe) : void

public : getGroupe() : Groupe

-Personne:

public : getNom() : string

public : getPrenom() : string

public : getAge() : int

public : getDateDeNaissance() : string

public : setNom(string nom) : void

public : toString() : string

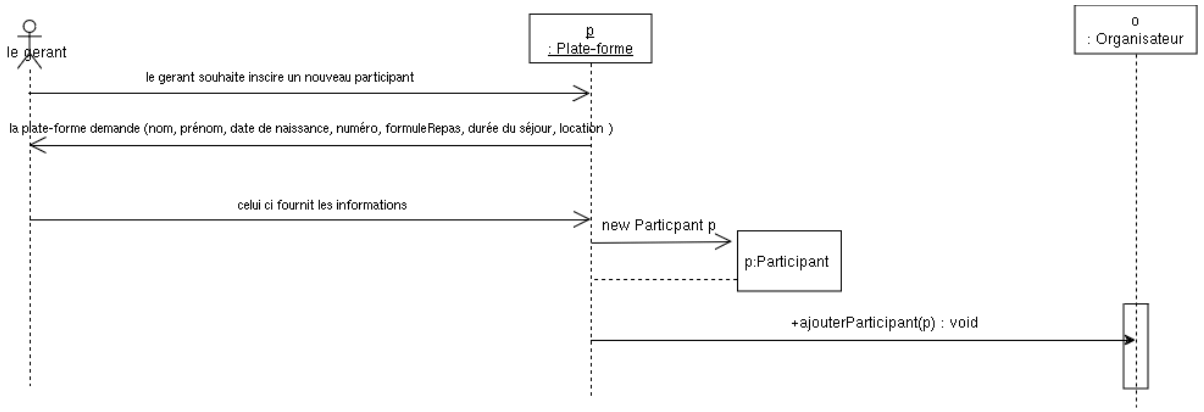
public : setPrenom(string prenom) : void

public : getNuméro() : string

public : setNumero(string numero) : void

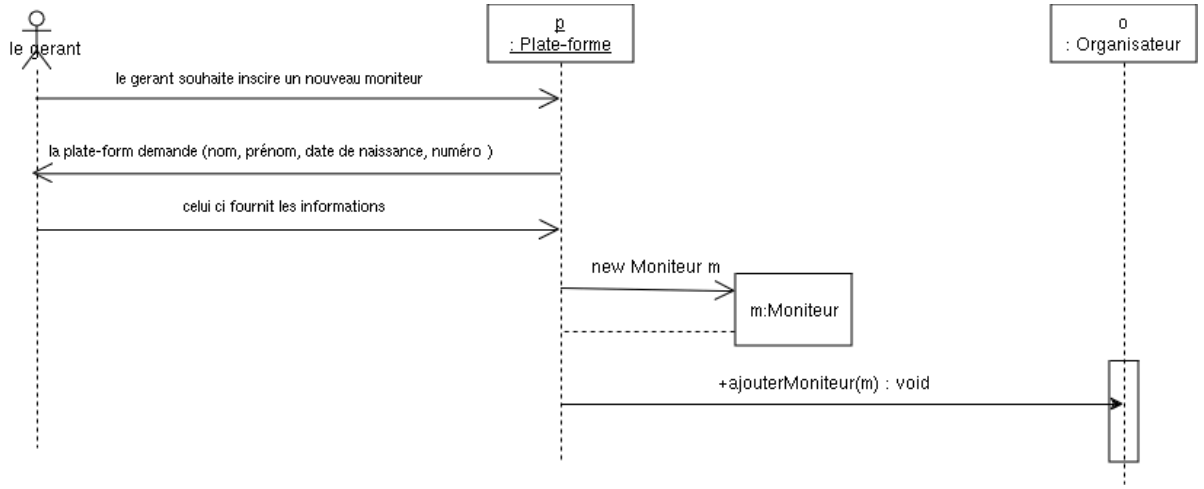
diagrammes de séquences :

UC1



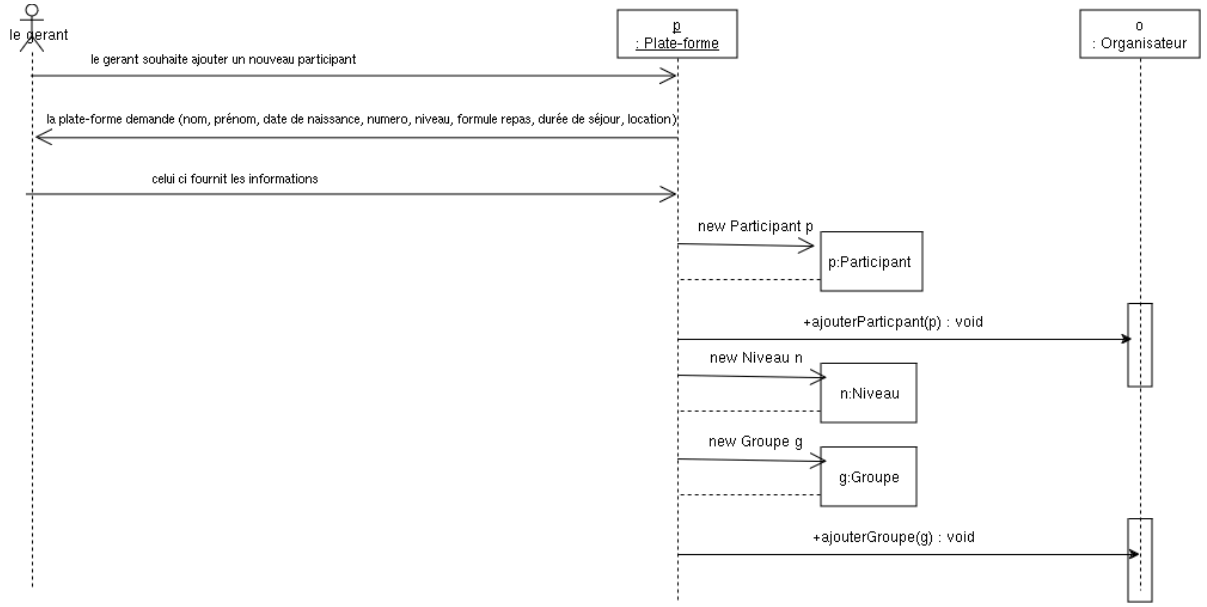
L'organisateur souhaite inscrire un nouveau participant,in interagit donc avec la plateforme. On lui demandera de saisir les informations concernant le participant (nom, prénom,date de naissance, numéro de téléphone, formule repas, durée du séjour, location). renseigne les informations. La plateforme crée un objet de type Participant grâce au constructeur de la classe puis à l'aide de la procédure ajouterParticipant() créée le profil du participant.

UC2



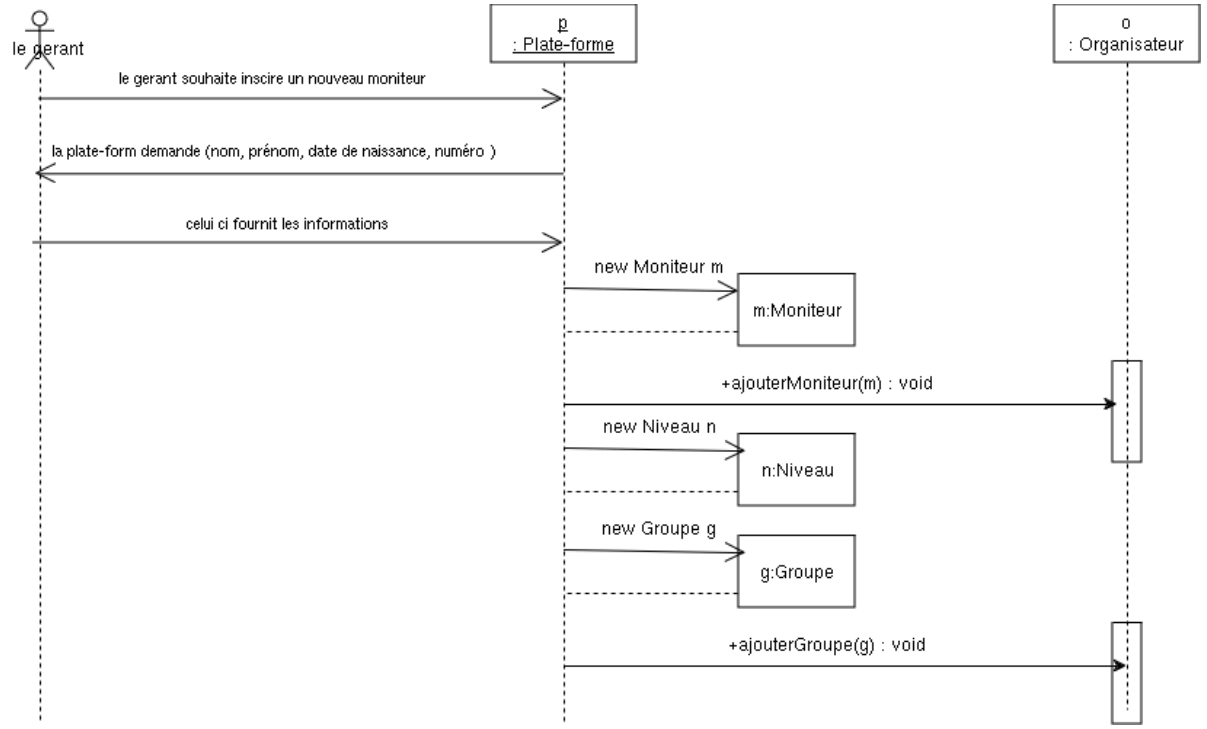
L'organisateur souhaite inscrire un nouveau moniteur, on interagit donc avec la plateforme en cliquant sur le bouton pour ajouter un nouveau moniteur. La plateforme lui demandera de saisir les informations concernant le nouveau moniteur (nom, prénom, date de naissance, numéro). L'organisateur va les saisir. Les informations seront récupérées pour créer un élément de type Moniteur et sera ajouté automatiquement à sa liste de moniteurs avec la méthode ajouterMoniteur().

UC3



L'organisateur souhaite ajouter un participant à un groupe,on interagit avec la plateforme en cliquant sur le bouton pour associer le participant. On considère que le participant et le groupe sont déjà crée. Il sélectionne le groupe auquel il veut ajouter le participant et le participant sera ajouté automatiquement à l'aide de la méthode ajouterGroupe().

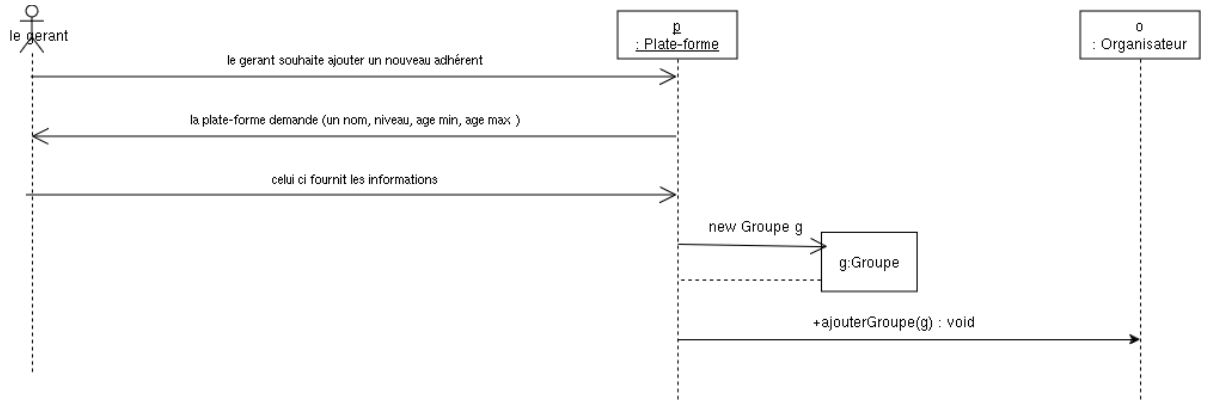
UC4



L'organisateur souhaite ajouter un moniteur à un groupe ,il interagit donc avec la plateforme. On lui demandera de saisir les informations concernant le moniteur (nom, prénom,date de naissance, numéro de téléphone). celui-ci renseigne les informations. On considère que le moniteur est déjà inscrit. La plateforme crée d'abord un objet de type niveau qui sera ensuite

attribué dans la création d'un objet groupe. L'organisateur peut désormais ajouter le moniteur au groupe.

UC5



L'organisateur souhaite créer un nouveau groupe, il va cliquer sur le bouton pour ajouter un groupe, on lui demande de saisir les informations pour ce groupe (libellé, niveau, âge minimum, âge maximum). L'organisateur saisit les informations. Les informations seront récupérées pour créer un nouvel élément de type Groupe et sera ajouté automatiquement à sa liste des groupes à l'aide de la procédure ajouterGroupe().